



Pelatihan Intel XDK

Modul 3. Pengenalan Intel XDK Development Tools.

Dikembangkan oleh Intel Software.



Versi 1.0. September 2013.

Hak Cipta (C) 2013 Intel Software.

Adobe, Adobe Edge, dan Adobe Edge Animate merupakan merk dagang dan merk dagang terdaftar milik Adobe, Inc.

Android, Chrome, dan Google adalah merk dagang dan merk dagang terdaftar milik Google, Inc.

Intel, Intel XDK merupakan merk dagang dan merk dagang terdaftar milik Intel Corporation.

iOS dan OS X adalah merk dagang dan merk dagang terdaftar milik Apple, Inc.

Windows, Windows Phone, Windows Store adalah merk dagang dan merk dagang terdaftar milik Microsoft Corporation.

Semua merk dagang terdaftar lainnya yang tercantum pada modul ini merupakan milik pemiliknya masing-masing yang peduli.

Modul ini bebas untuk disebarluaskan secara hardcopy ataupun softcopy dengan tanpa menambah atau mengurangi isi dari modul ini termasuk isi halaman ini. Modul ini tidak boleh diperjual belikan atau digunakan untuk tujuan komersial lain.

Muhammad Yusuf

muhammad.yusuf@live.com

@ruangchupa

Gilang Abdul Aziz

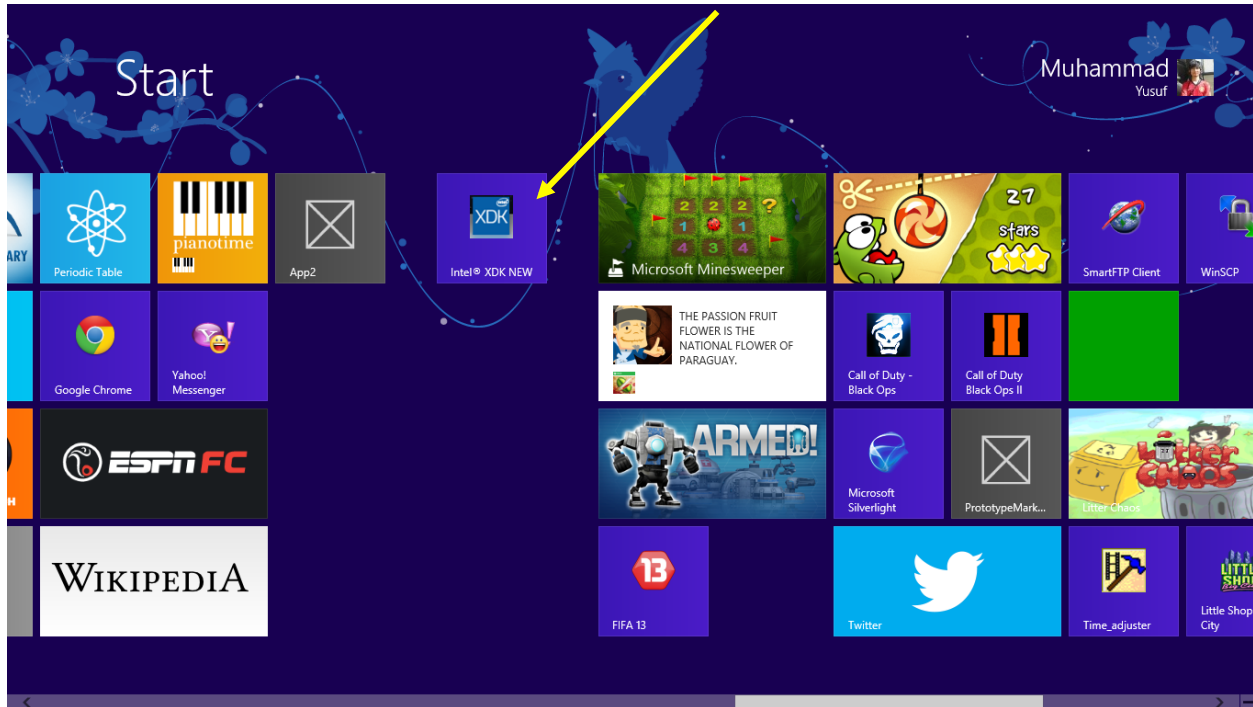
gilang@llovvia.web.id

@Ltheordinary

Modul 3. Pengenalan Intel XDK Development Tools.

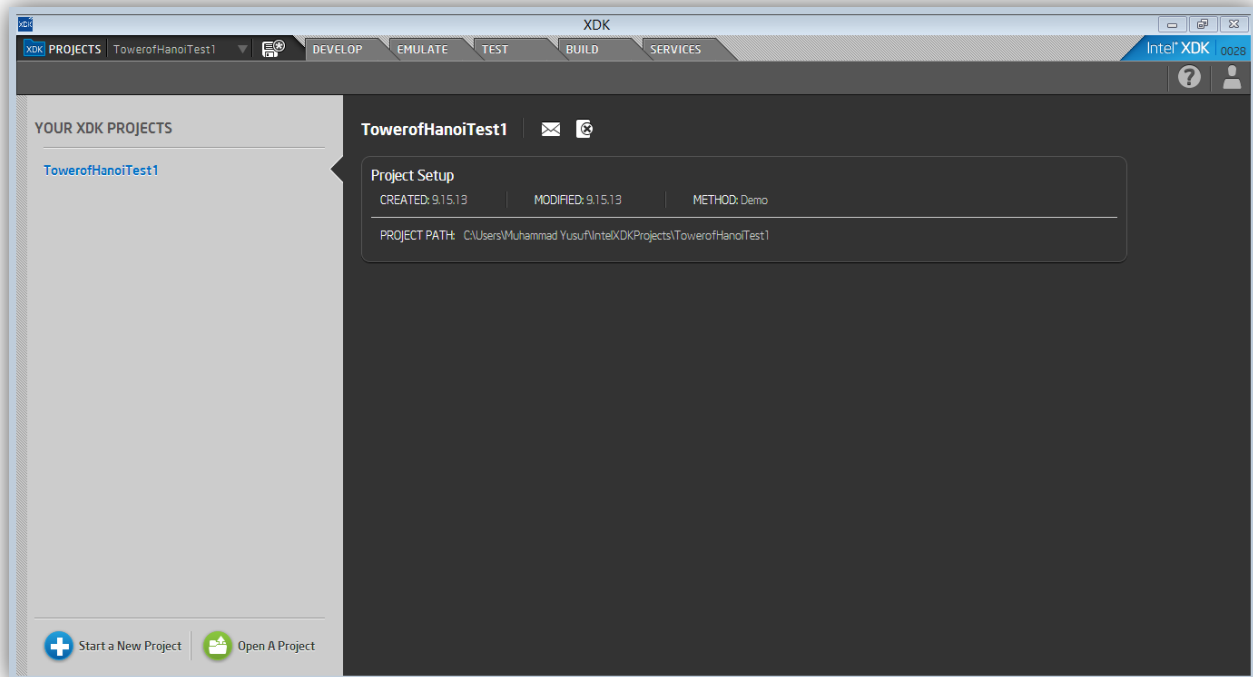
Membuka aplikasi Intel XDK

Setelah diinstall, aplikasi Intel XDK dapat dibuka dengan mengklik shortcut yang terdapat pada start menu di komputer Anda dengan nama aplikasi **Intel® XDK New**. Sebagai contoh berikut adalah shortcut Intel XDK pada start menu Windows 8:

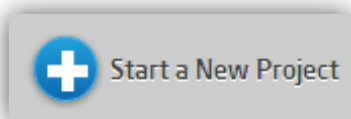


Tampilan Awal Intel XDK

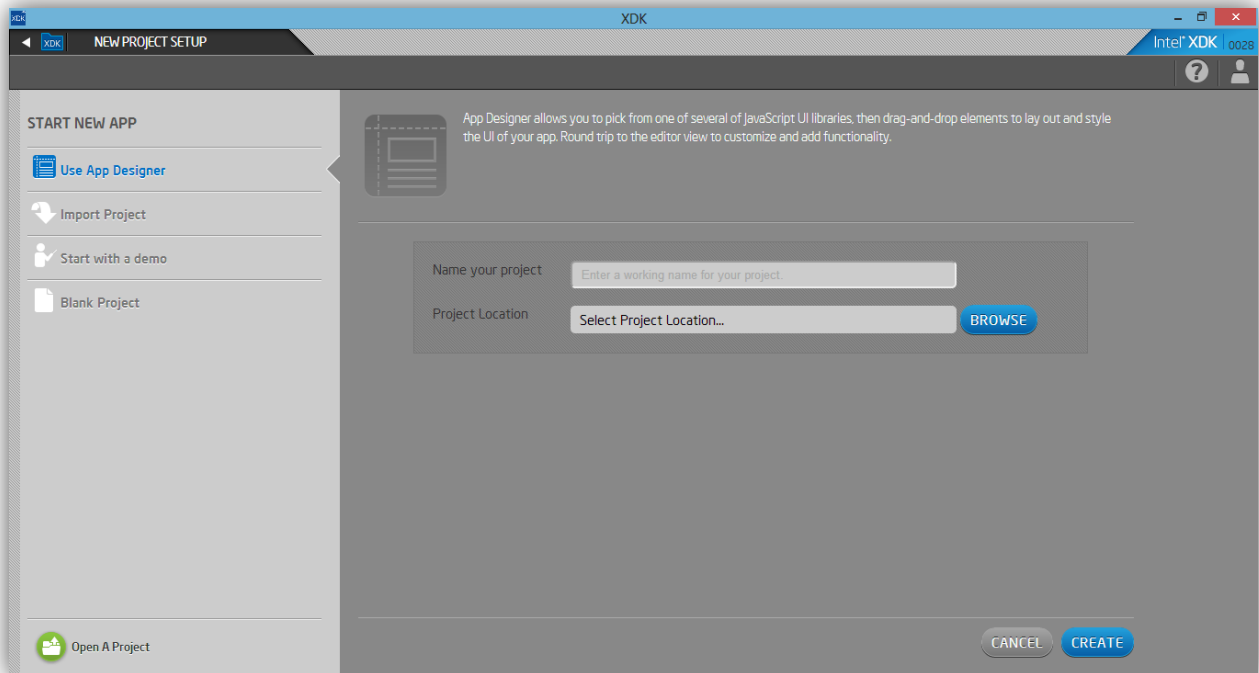
Pada saat pertama dibuka, maka akan muncul tampilan awal sebagai berikut:



Pada tampilan tersebut disajikan menu untuk membuka proyek-proyek yang telah kita buat sebelumnya dan menu untuk membuat proyek baru. Kali ini kita akan mencoba membuat satu proyek baru. Silakan klik tombol Start a New Project.



Sekarang kita akan disuguhkan tampilan menu untuk membuat proyek baru.



Pada bilah di samping kiri, terdapat pilihan tipe metode dari pembuatan proyek yang akan kita buat. Terdapat 4 pilihan yaitu **Use App Designer**, **Import Project**, **Start with a demo**, dan **Blank Project**:

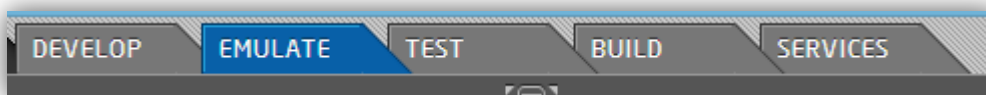
1. **Use App Designer**: Dengan menggunakan metode pembuatan proyek ini, kita akan diberikan beberapa library JavaScript UI yang dapat kita pilih untuk membuat tampilan pada aplikasi kita. Kita juga diberikan kemudahan dengan cara drag and drop saat menambahkan dan melakukan layouting elemen pada aplikasi kita.
2. **Import Project**: Metode pembuatan proyek ini digunakan saat kita ingin mengimport proyek yang telah kita buat sebelumnya menggunakan Intel XDK versi sebelumnya, App Starter Wizard, atau lainnya yang kompatibel dengan Intel XDK.
3. **Start with a demo**: Metode pembuatan proyek yang menggunakan aplikasi-aplikasi demo yang sudah disediakan oleh Intel sebagai contoh.
4. **Blank Project**: Metode pembuatan proyek ini dipakai jika kita ingin membuat proyek yang benar-benar dari awal dan saat awal dibuat hanya berisi direktori standar Intel XDK dan satu file HTML awal. Namun saat ditengah-tengah pengembangan proyek jika ingin menggunakan App Designer, tetap dapat menggunakannya dengan mengklik pada menu **App Designer** yang sudah disediakan.

Silakan pilih **Start with a Demo**, lalu pilih salah satu demo aplikasi yang disediakan pada menu sebelah kanan dan klik **Next**. Pada tampilan berikutnya, silakan beri nama proyek dan pilih lokasi folder penyimpanan proyek dan klik Create.

Sebenarnya ada satu lagi metode pembuatan proyek yang ditawarkan Intel XDK, yaitu metode App Starter Wizard. Dengan metode ini kita dapat membuat sebuah aplikasi sampai jadi dengan beberapa klik saja. Namun saat modul ini ditulis, metode App Starter Wizard belum dimasukkan pada versi Intel XDK New. Intel berjanji akan segera mengembalikan metode pembuatan proyek ini ke versi Intel XDK New. Namun kami telah menyediakan modul yang membahas tentang metode pembuatan proyek menggunakan App Starter Wizard ini pada modul ekstra.

Sekilas Tahapan Pengembangan pada Intel XDK

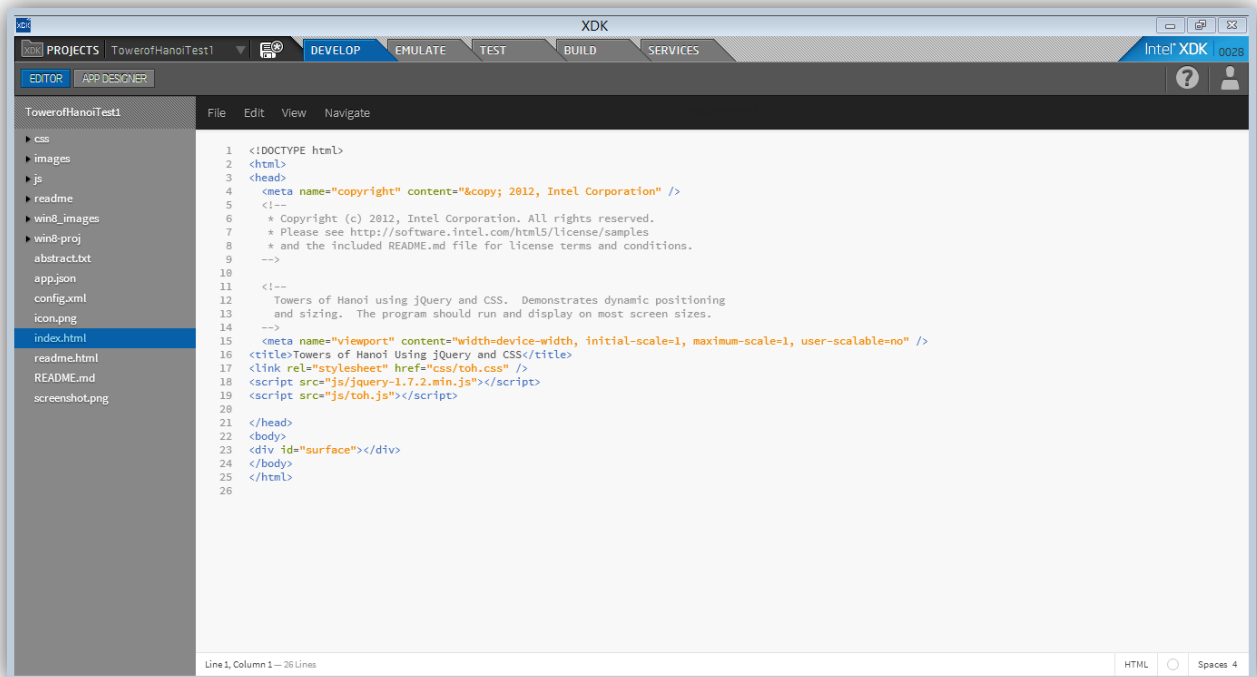
Jika kita melihat bagian sebelah atas dari aplikasi Intel XDK, maka akan terdapat menu seperti pada gambar di bawah ini:



Masing-masing menu tersebut merepresentasikan tahapan pengembangan pada Intel XDK yaitu sebagai berikut:

1. Develop

Tahapan ini adalah tahap saat aplikasi dibuat dan dikembangkan. Terdapat dua pilihan pengembangan pada tahap ini yaitu **Editor** dan **App Designer**. Pada pilihan Editor kita dapat melakukan aktivitas Coding dan pada pilihan App Designer kita dapat melakukan aktivitas layouting.



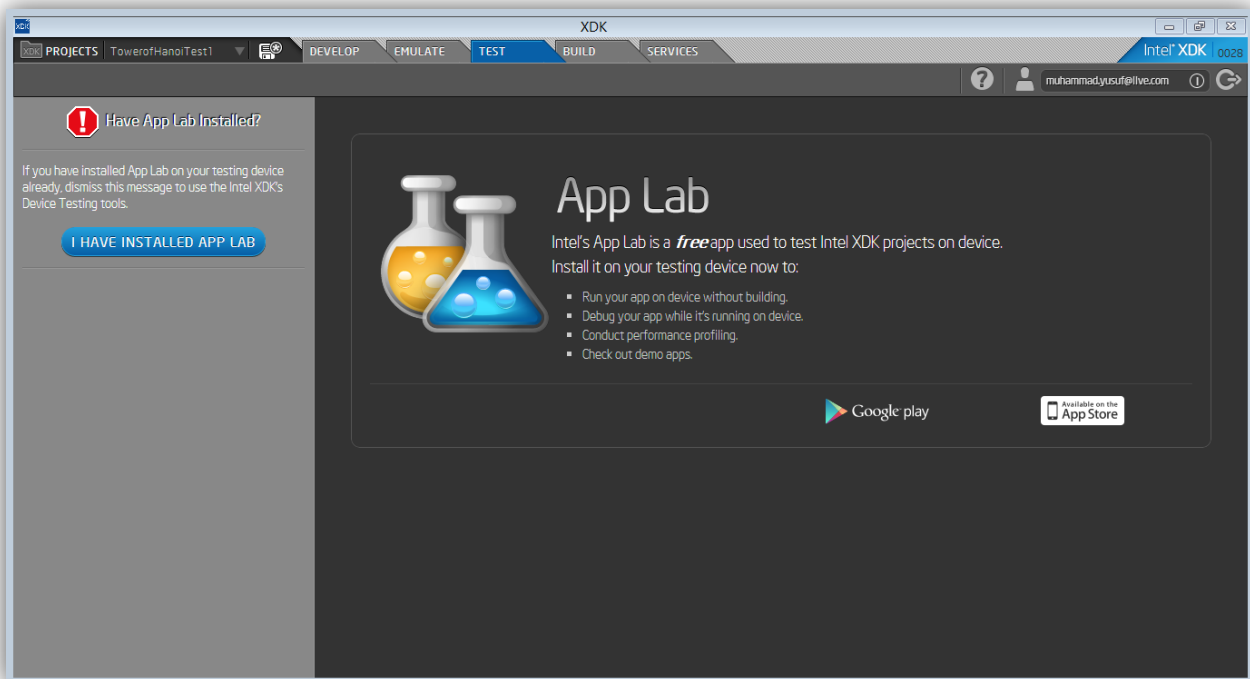
2. Emulate

Pada tahap ini aplikasi yang kita buat dites untuk disimulasikan ketika berjalan pada device nanti. Terdapat banyak menu untuk melakukan simulasi seperti Accelerometer, Geolocation, Network, Events, dan lain-lain.



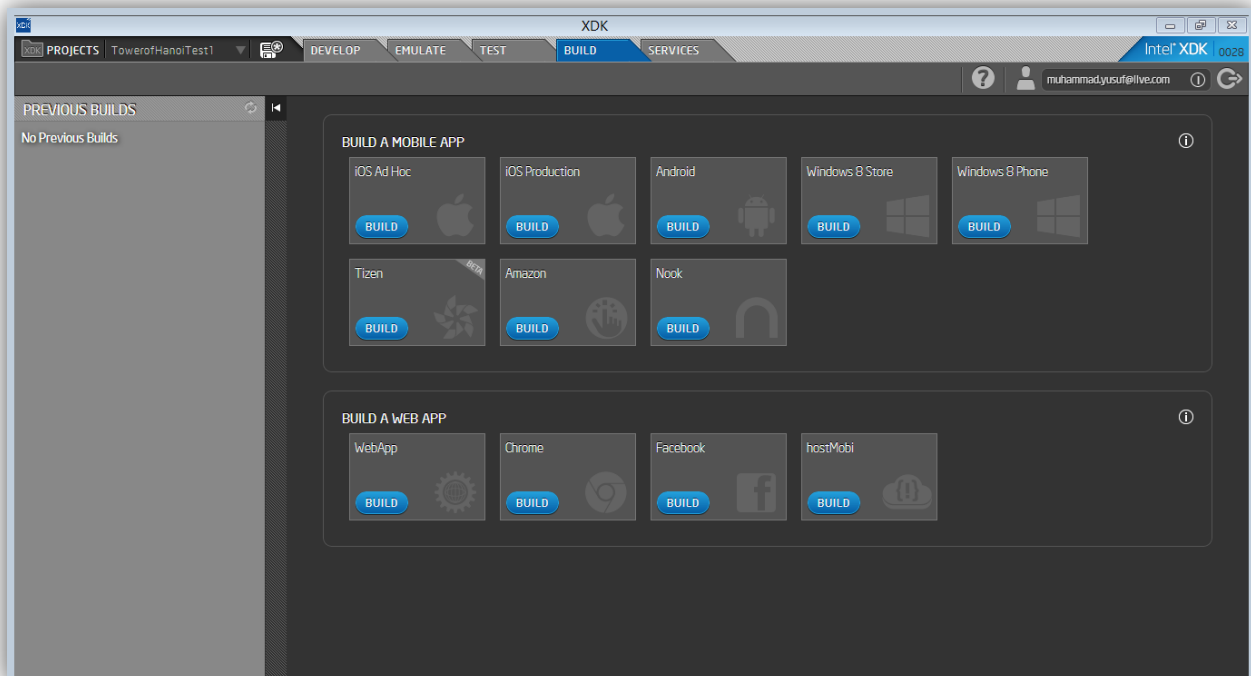
3. Test

Pada tahapan ini aplikasi yang kita buat akan dicoba untuk dijalankan di device asli. Yang perlu dilakukan untuk melakukan tahap ini adalah kita terlebih dahulu perlu menginstall aplikasi **App Lab** di device yang akan dijadikan tempat kita mencoba aplikasi kita. Nanti, pada aplikasi App Lab tersebut kita dapat membuka aplikasi yang kita kembangkan yang telah kita push ke server pada saat tahap Test ini. Aplikasi App Lab ini tersedia untuk device dengan platform iOS dan Android.



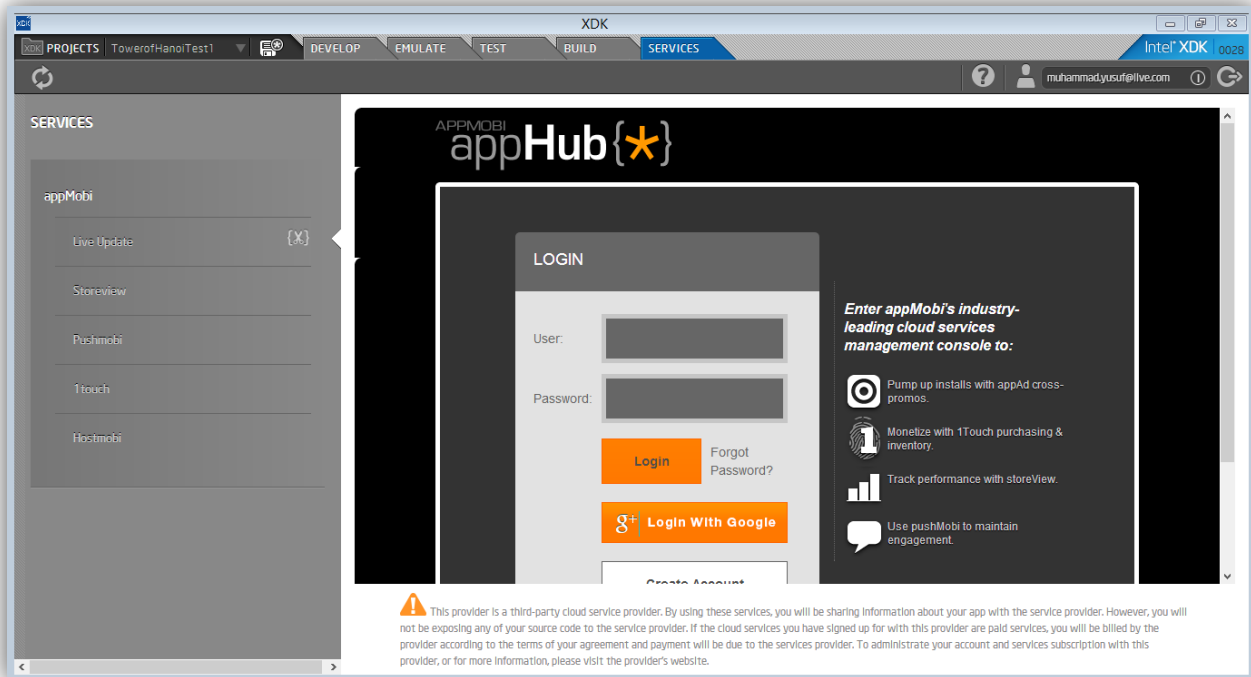
4. Build

Tahap ini adalah tahap saat aplikasi kita dibuild menjadi sebuah aplikasi untuk masing-masing platform yang akan kita pilih. Caranya cukup mudah yaitu hanya dengan mengklik salah satu platform pada menu di tahap Build ini maka aplikasi kita akan diunggah ke server dan oleh server akan dikembalikan lagi dalam bentuk paket aplikasi yang siap diinstall atau dipublish di platform yang telah kita pilih.



5. Services

Tahap ini sebenarnya adalah tahap tambahan yang optional jika kita ingin menggunakan layanan-layanan yang disediakan untuk melakukan hal misalnya menambah iklan di aplikasi kita, men-track performance aplikasi kita, dan lain-lain.

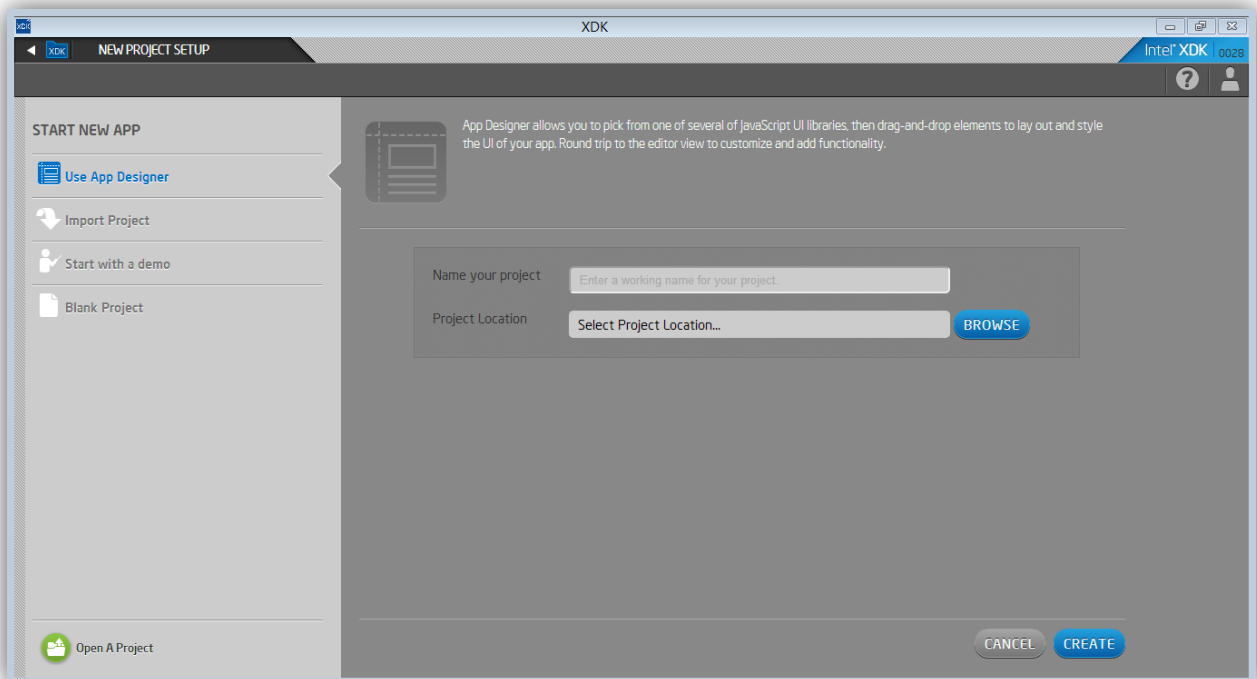


Membuat Aplikasi Hello World pada Intel XDK

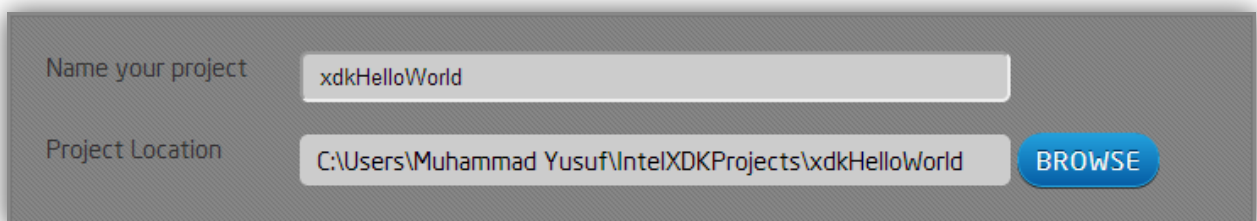
Kali ini, untuk mencoba beberapa fitur dan tool pada Intel XDK mari kita membuat sebuah proyek aplikasi simpel yaitu aplikasi Hello World. Mari membuat sebuah proyek baru menggunakan App Designer.

1. Membuat Proyek

Pada tampilan awal Intel XDK, pilih **Start a New Project** dan pilih **Use App Designer** sebagai metode pembuatan proyek yang akan kita pakai.



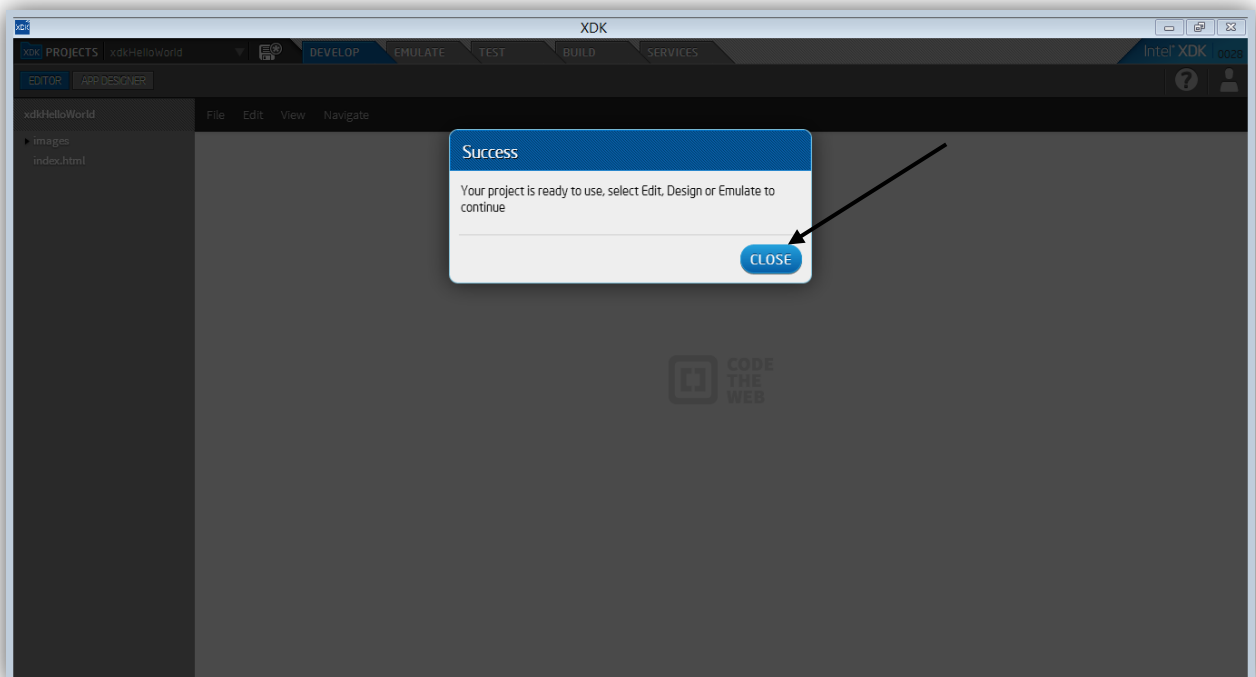
Beri nama proyek yang akan kita buat dengan nama **xdkHelloWorld** dan silakan tentukan lokasi folder penyimpanan proyek tersebut.



Klik tombol **Create** untuk memulai proses pembuatan proyek.

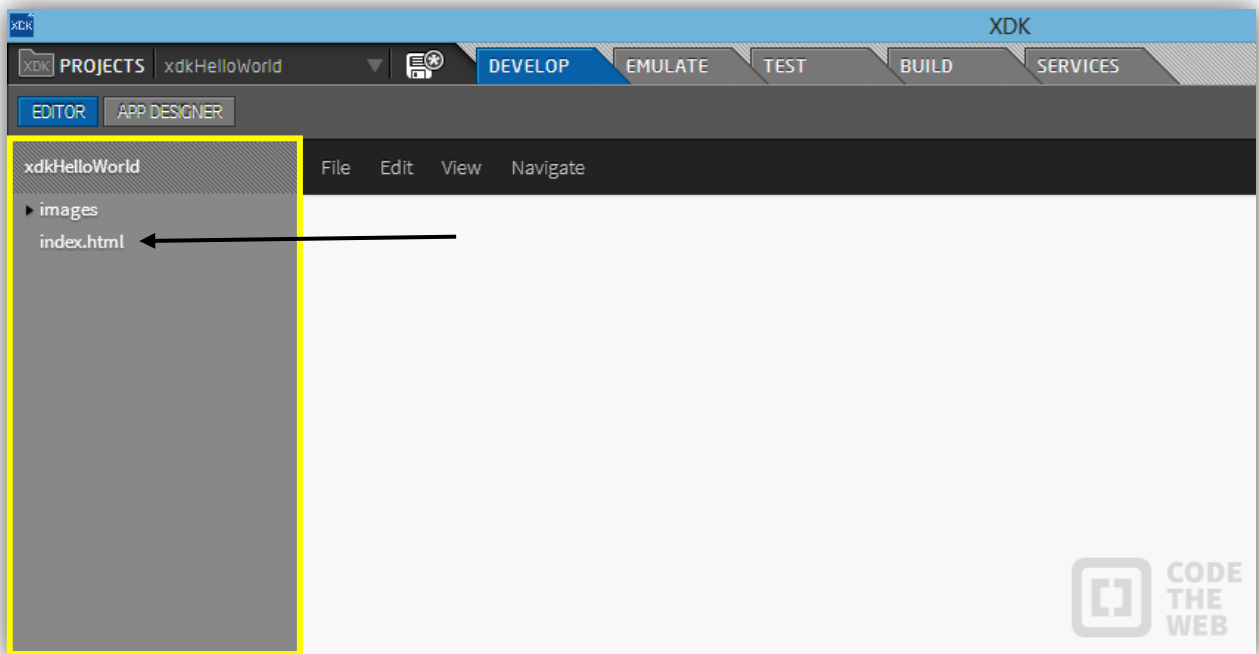


Setelah proyek kita selesai dibuatkan oleh Intel XDK, maka akan muncul tampilan bahwa proyek kita telah sukses dibuat dan siap untuk dikembangkan lebih lanjut. Silakan tutup pop up **Success** yang muncul.

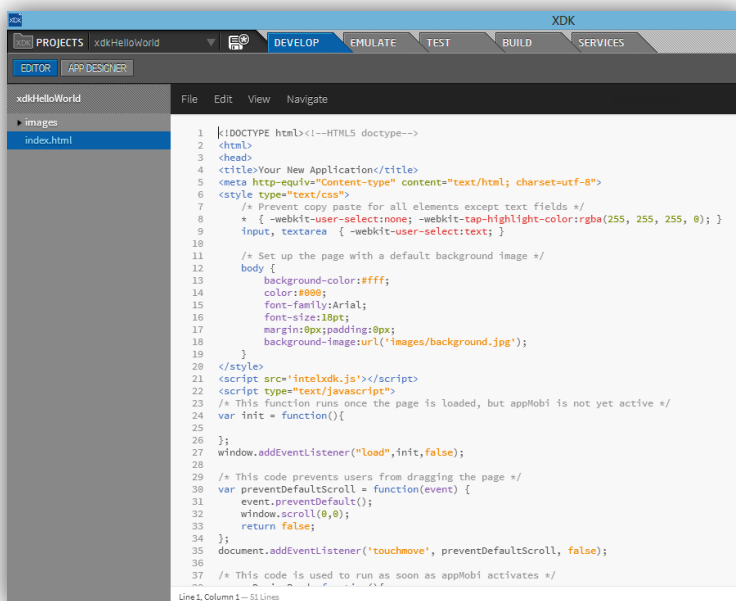


2. Menambahkan elemen-elemen aplikasi

Sekarang, saatnya mulai bekerja. Pada bilah sebelah kiri terlihat hierarki file dari proyek kita. Silakan klik file **index.html**.



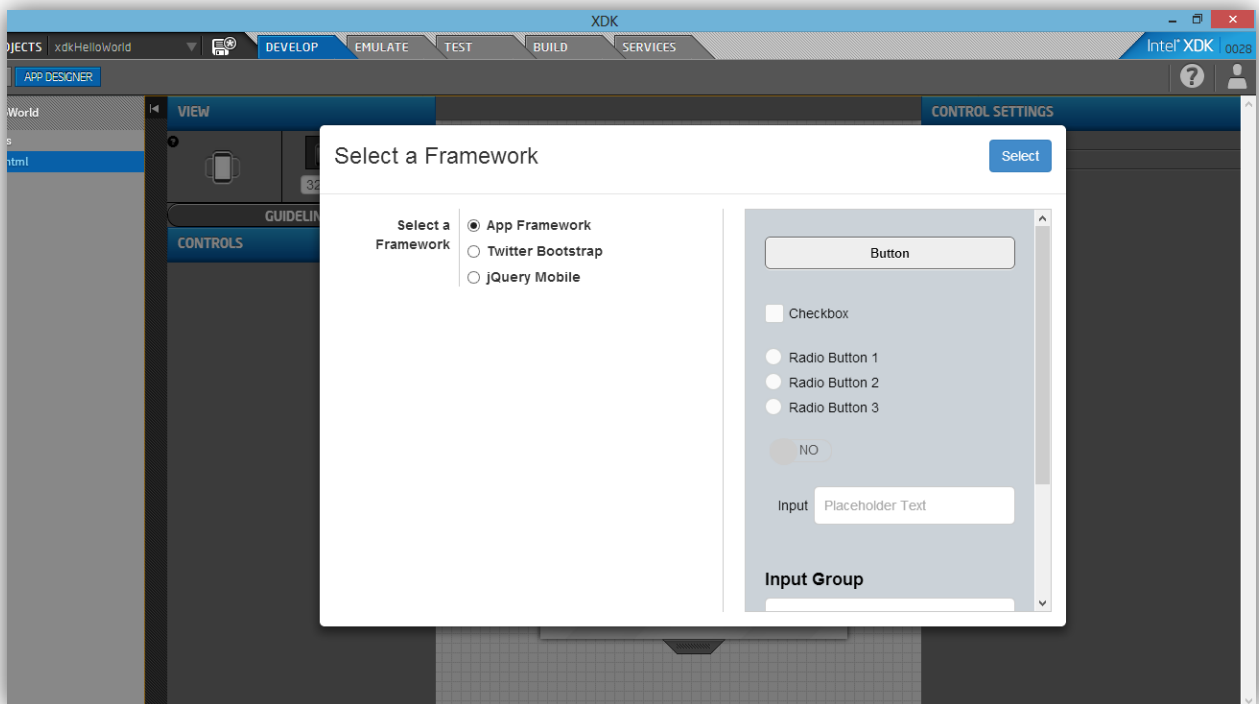
Maka akan terbuka pada bilah **Editor** tampilan kode program dari file index.html tersebut.



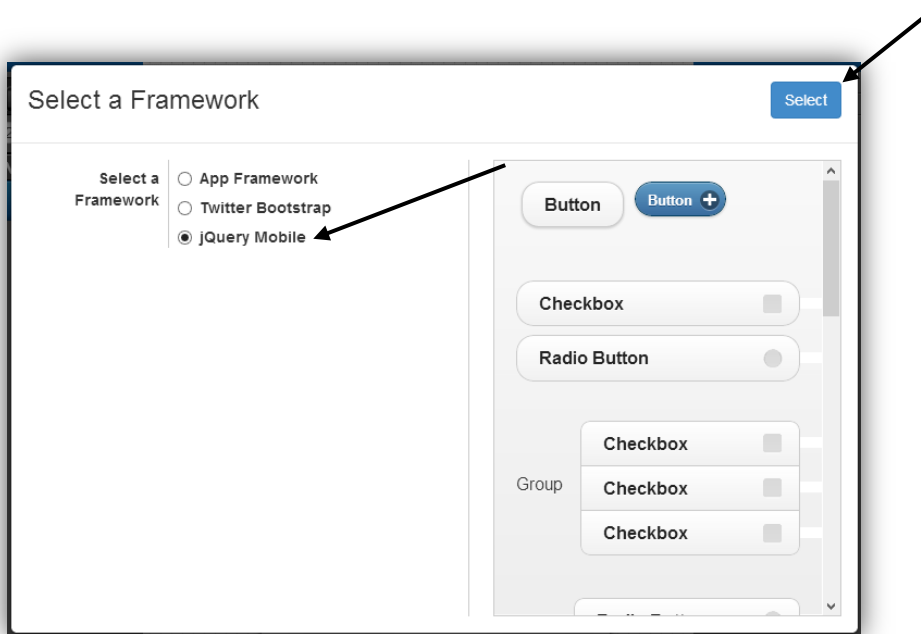
Lalu klik pada tombol **App Designer** di bagian atas bilah sebelah kiri untuk membuka tampilan bilah App Designer.



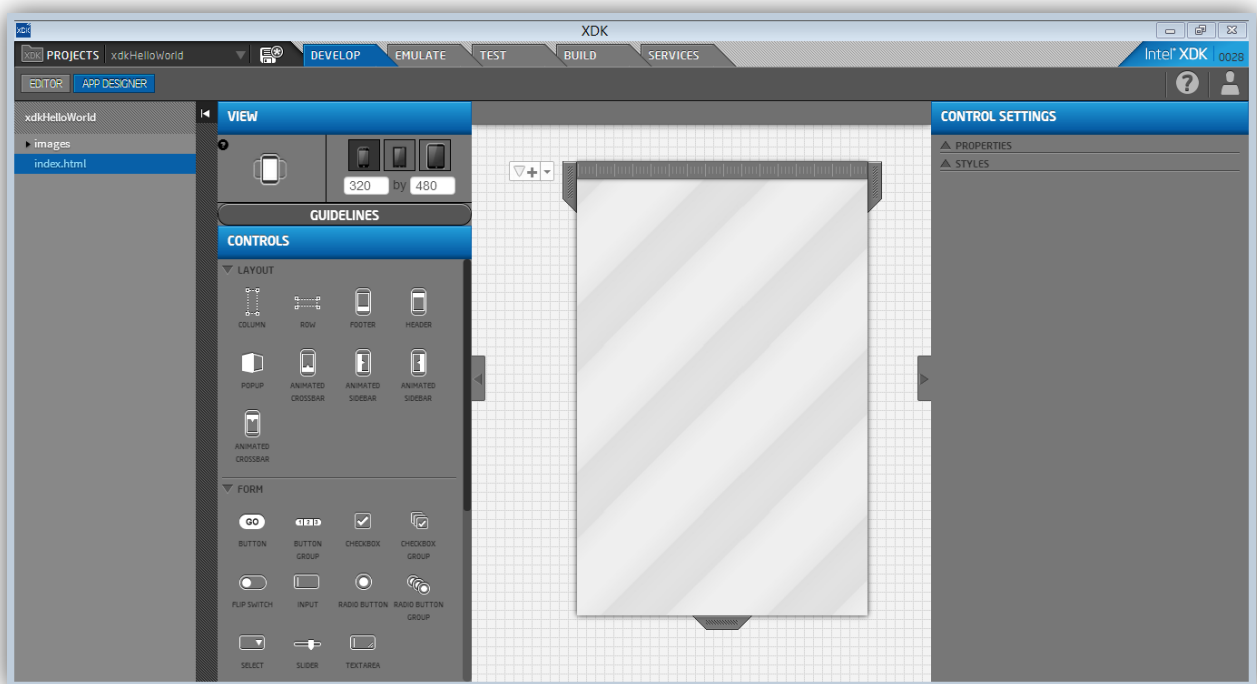
Maka bilah Editor kita akan berubah menjadi bilah App Designer dan untuk pertama kali, kita akan diminta memilih framework UI yang mana yang akan kita pakai.



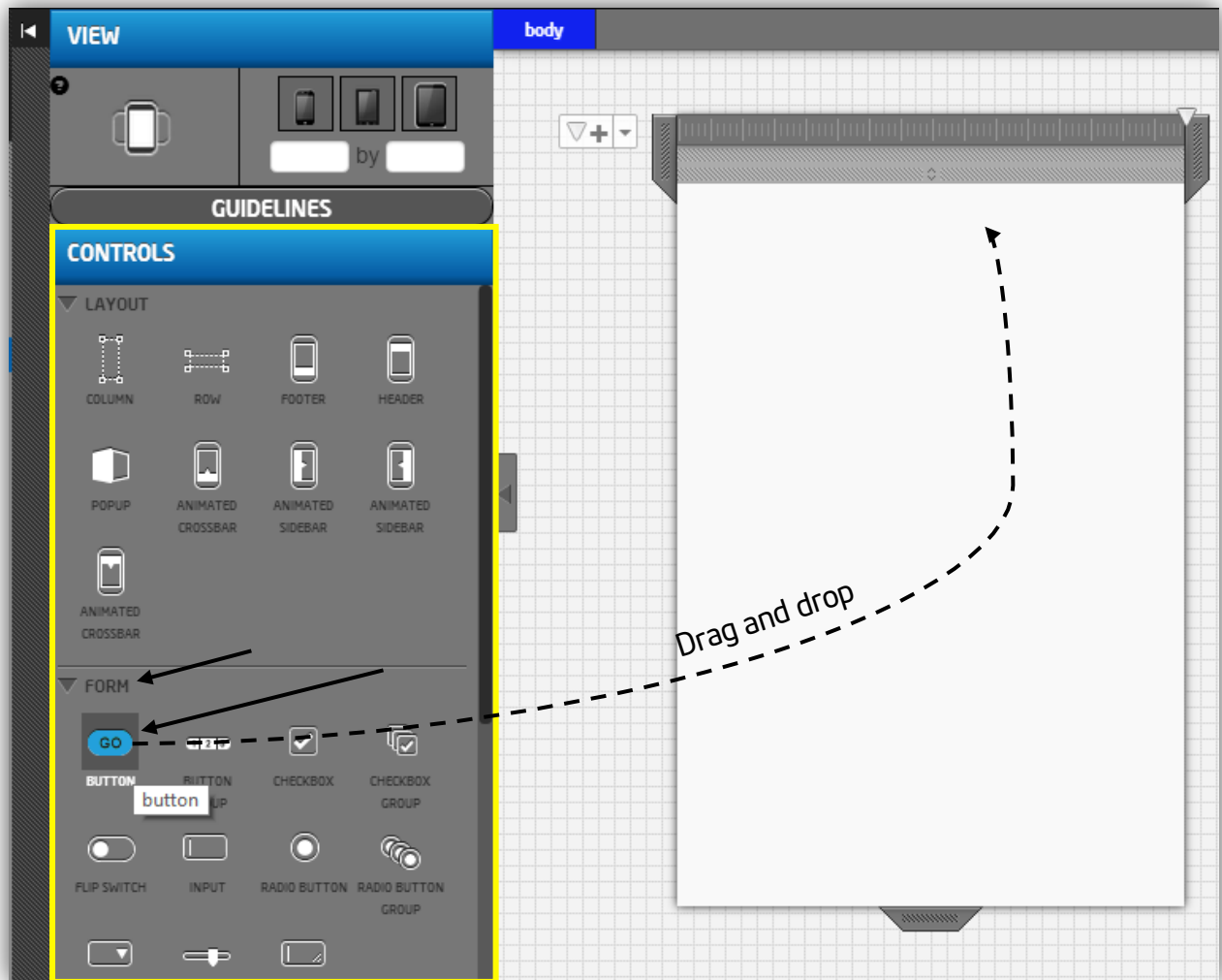
Pilih framework **jQuery Mobile** dan klik tombol **Select**.



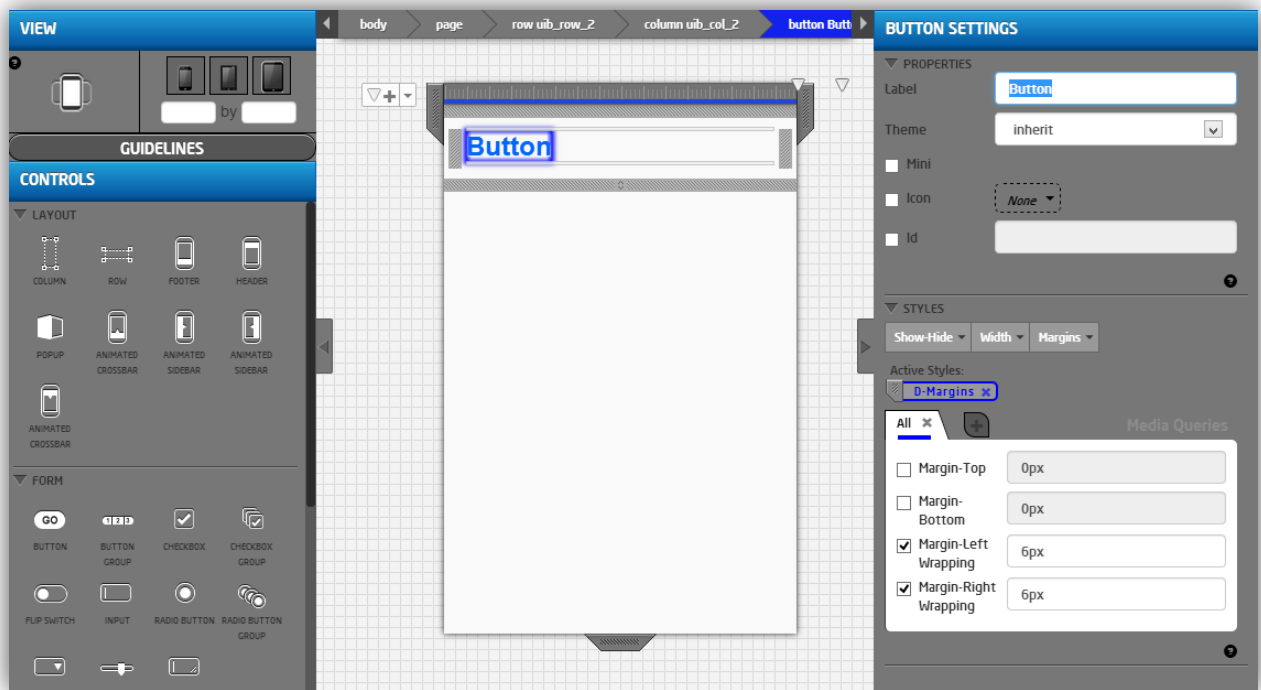
Maka tampilan khas App Designer akan muncul beserta kanvas aplikasi kita dan kita bisa dengan bebas melakukan aktivitas drag and drop untuk menambahkan elemen dan me-layout elemen pada aplikasi kita.



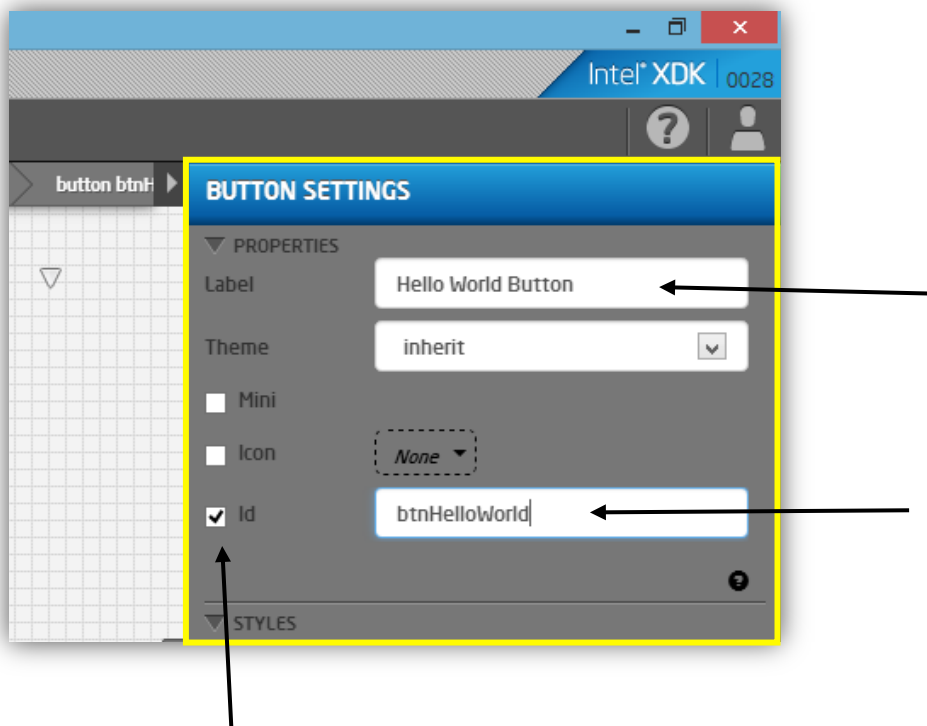
Pada bilah **Controls**, pada bagian **Form**, silakan drag ke kanvas sebuah elemen **Button**.



Sehingga tampilan di kanvas akan menjadi seperti ini:

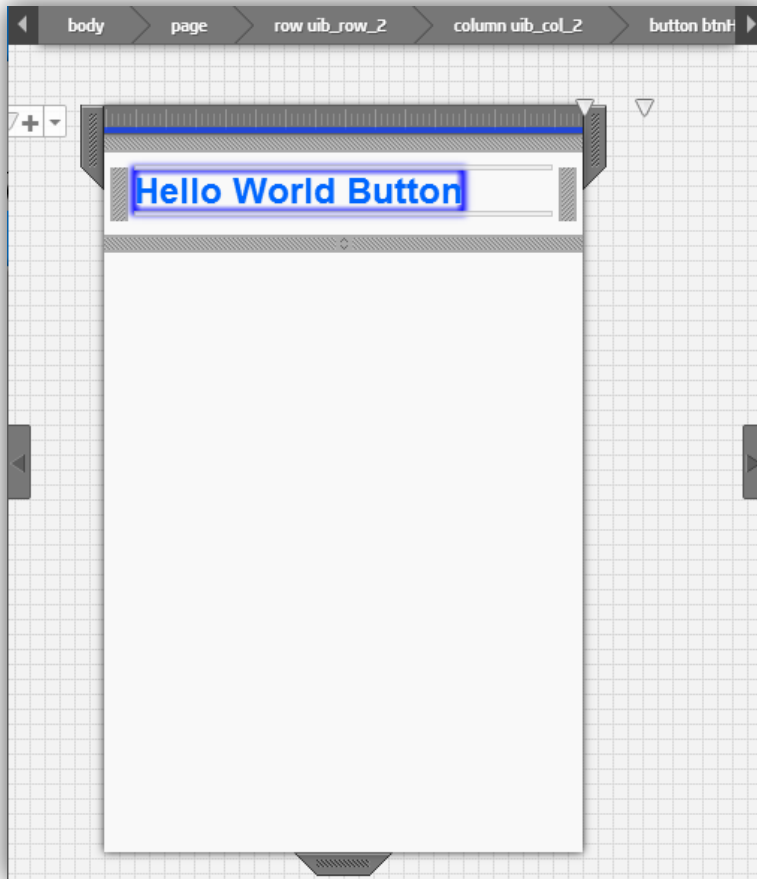


Dengan button yang baru kita buat yang masih terseleksi, pada bilah **Button Settings**, isi **Label** dengan "Hello World Button", lalu centang **Id** dan isi dengan "btnHelloWorld".



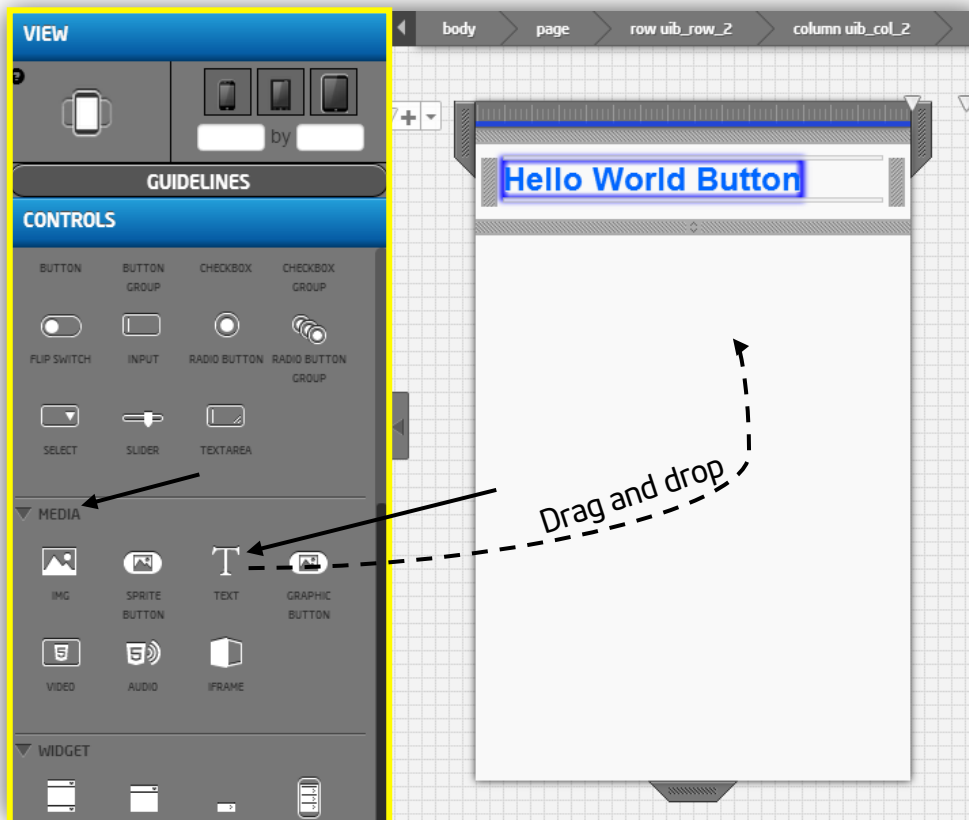
Pengisian **label** adalah untuk merubah teks pada tombol tersebut di tampilan layar dan pengisian **id** adalah untuk mendefinisikan id dari tombol tersebut untuk penggunaan di kode program nanti.

Tampilan di kanvas kita sekarang adalah seperti ini:

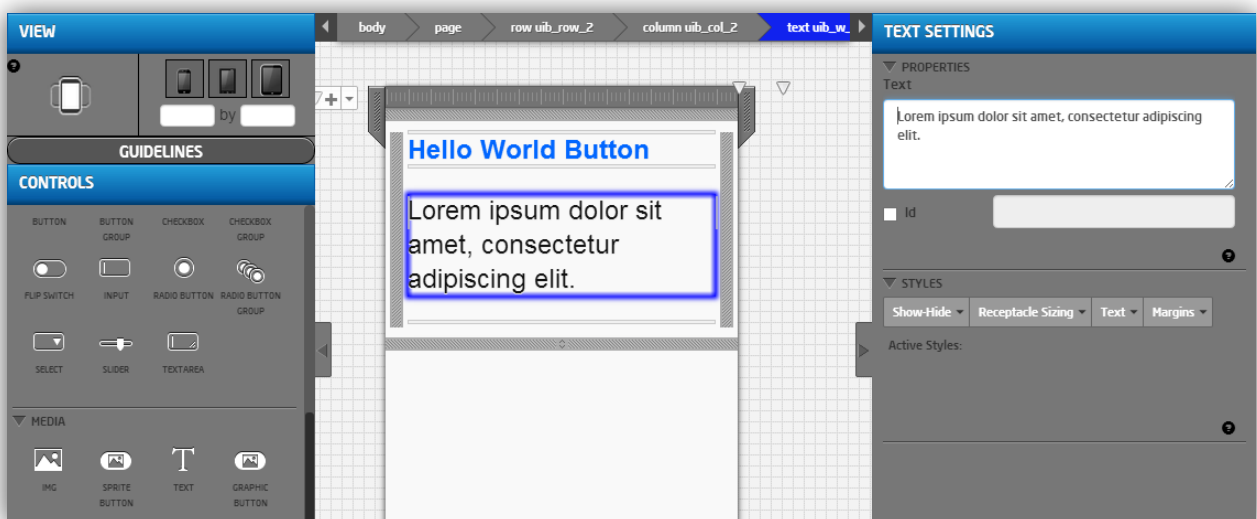


Pemberian nama Id haruslah unik (berbeda-beda) pada setiap elemen yang kita buat. Sama sekali tidak boleh ada kesamaan. Perlu diingat juga bahwa Id bersifat case-sensitive.

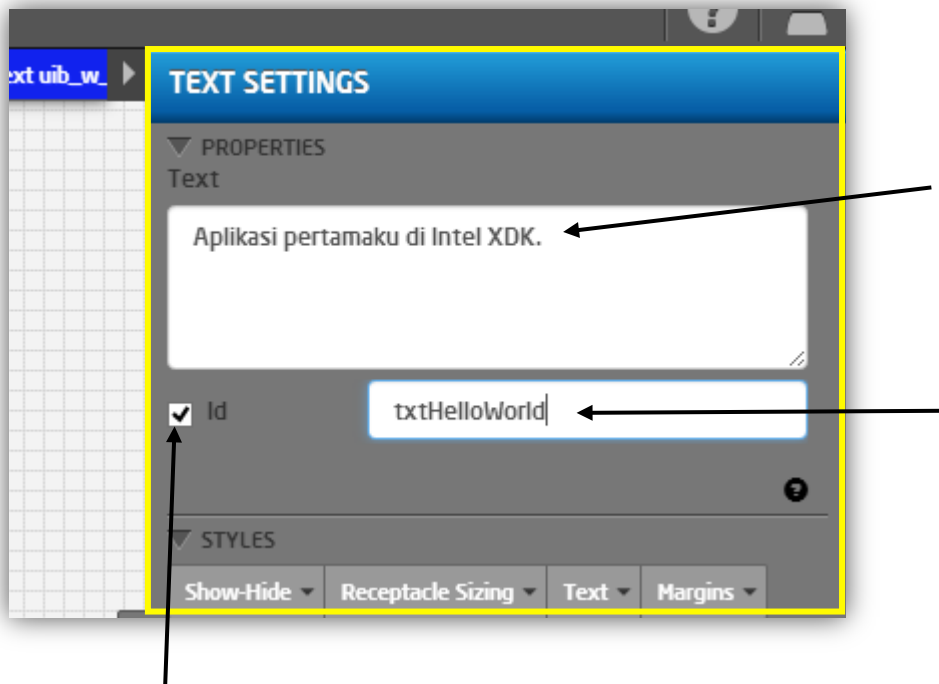
Berikutnya kita akan menambahkan sebuah teks di kanvas kita. Pada bilah **Controls**, bagian **Media**, drag and drop **Text** ke sebelah bawah tombol **btnHelloWorld** yang telah kita buat sebelumnya.



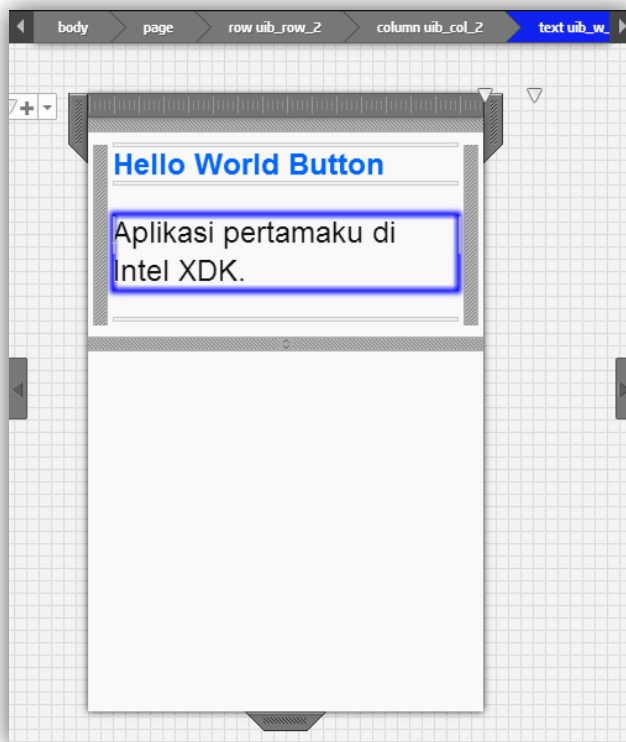
Sehingga tampilan di kanvas akan seperti berikut ini:



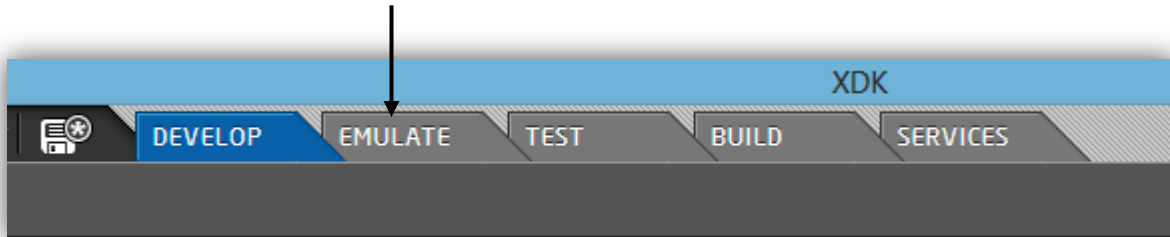
Lalu masih dengan teks tersebut yang terseleksi, pada bilah Text Settings, isi Text dengan "Aplikasi Pertamaku di Intel XDK". Lalu centang Id, dan isi dengan "txtHelloWorld".



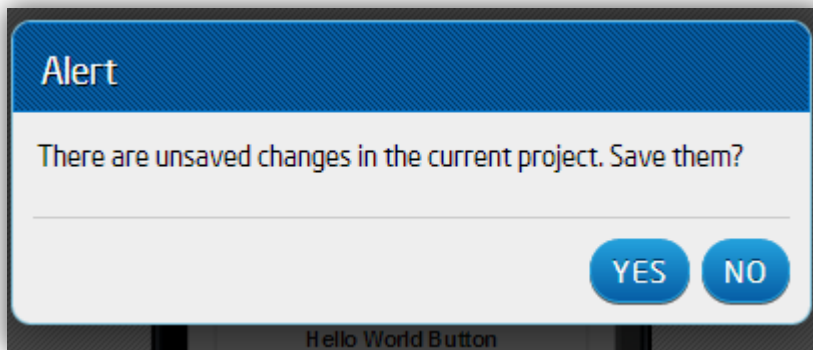
Maka tampilan di kanvas kita akan seperti berikut:



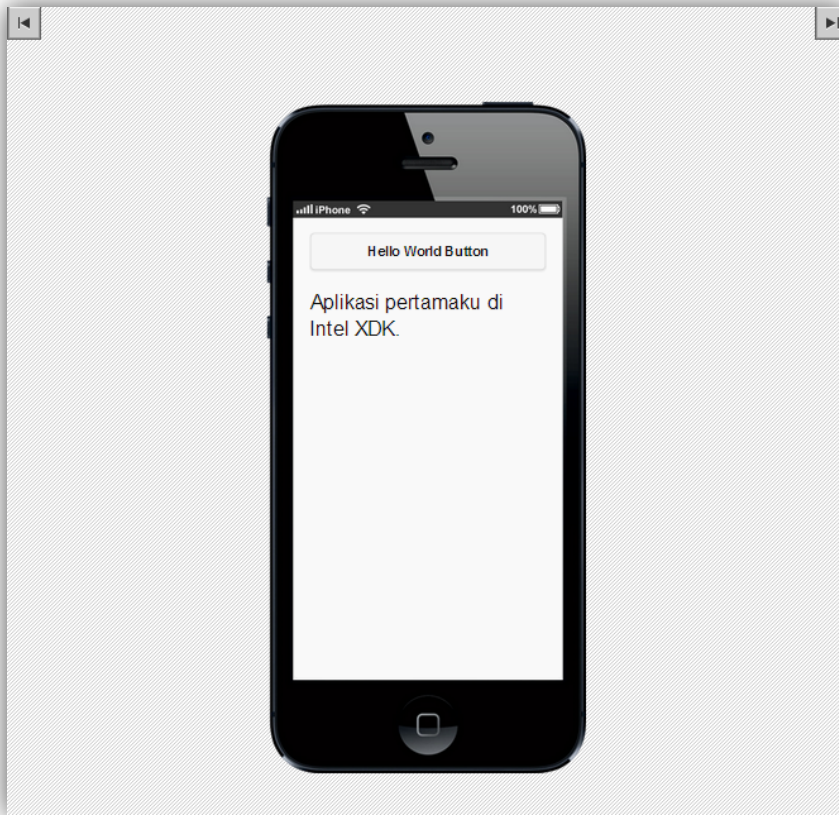
Penambahan dan layouting elemen pada aplikasi kita sudah cukup, sekarang akan kita lanjutkan pada coding alias meng-kode program. Sebelumnya, mari kita coba tes aplikasi kita dengan mengklik tombol **Emulate** pada menu tahapan-tahapan pengembangan di bagian atas.



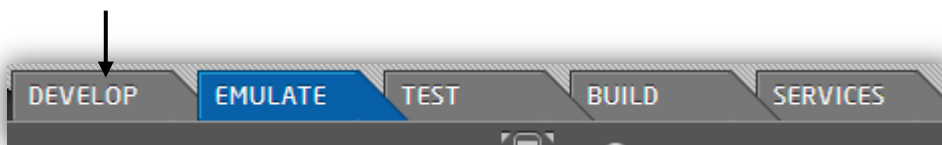
Saat tombol tersebut diklik, kemungkinan akan ada alert permintaan untuk men-save aplikasi kita jika belum kita save dan silakan pilih **Yes**.



Dan berikut adalah tampilan aplikasi kita yang sedang diemulasikan pada device virtual, untuk contoh di bawah ini adalah device iPhone 5 virtual:



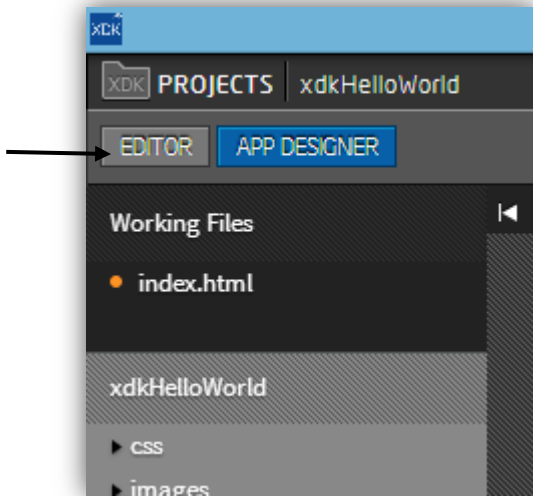
Lumayan bukan? Jika kita klik tombol Hello World Button yang telah kita buat maka tidak akan terjadi apa-apa. Mengapa? Jelas karena kita belum memberi perintah aksi apapun pada tombol tersebut. Maka dari itu, mari kita kembali ke tampilan Develop dengan mengklik tombol **Develop** pada menu tahapan-tahapan pengembangan di bagian atas.



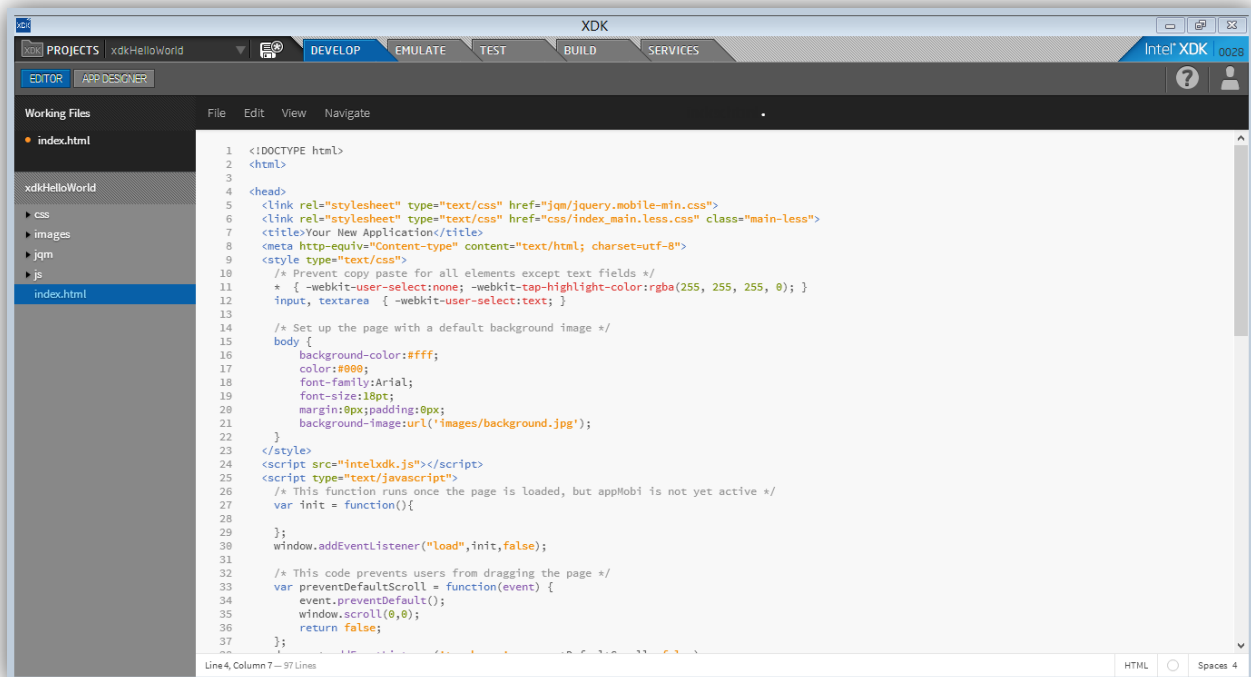
Sekarang, saatnya kita melakukan aktivitas mengetik. Siap untuk mengetik? ☺

3. Menambahkan kode untuk aksi program

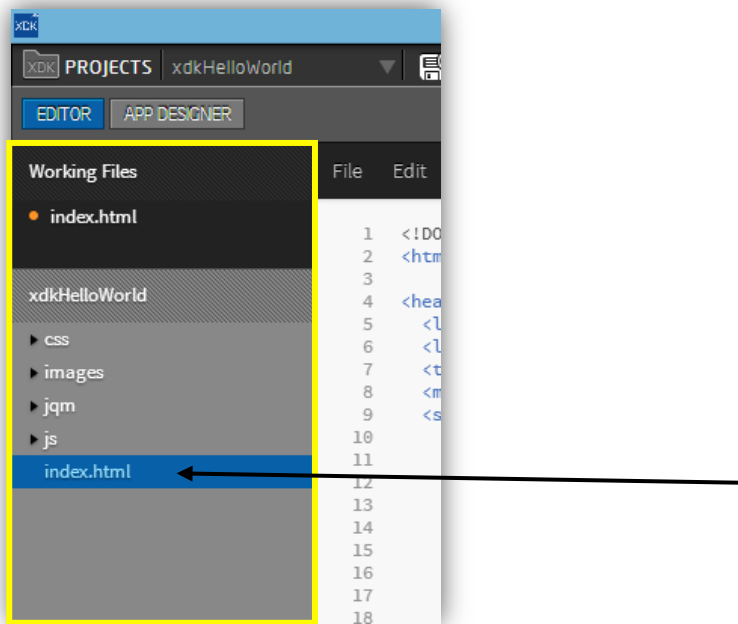
Pertama-tama, pada bagian sebelah kiri atas, pada menu di sebelah atas bilah hierarki file kita, klik tombol **Editor**.



Maka kita akan diberikan tampilan editor kode program yang penuh dengan huruf-huruf dan karakter-karakter seperti berikut ini:



Pada bilah **hierarki file**, pastikan file yang terpilih adalah file **index.html** untuk memastikan bahwa kode program yang sedang terbuka adalah kode program milik **index.html**.



Selanjutnya, mari fokus ke editor kode program yang terbuka, dan mari mulai aktivitas coding.

Bahasa yang akan kita gunakan adalah bahasa HTML, terlihat dari ekstensi file yang sedang kita buka, yaitu .html. Mungkin kebanyakan dari Anda sudah familiar dengan bahasa HTML ini. Bagi yang belum familiar, dapat belajar lebih lanjut tentang bahasa HTML pada modul yang khusus membahas tentang bahasa HTML. Modul tersebut banyak tersedia di HTML, buku yang membahas khusus tentang bahasa HTML pun banyak tersedia di toko buku.

Ya, serius, kita akan membuat aplikasi mobile menggunakan bahasa HTML yang lumayan familiar bagi Anda. Benar-benar bahasa HTML. Tambahannya mungkin ada pada kode Javascript yang akan kita tambahkan dan kode dari library-library tambahan. Untuk mengetahui betapa mudahnya, mari kita mulai.

Pertama-tama, mari kita lihat hasil pekerjaan kita sebelumnya saat menambahkan elemen pada kanvas. Aktivitas drag and drop tersebut sebenarnya akan otomatis men-generate kode HTML. Silakan scroll kode program Anda sampai Anda temukan kode seperti berikut:

```
<div class="widget-container content-area vertical-col">
  <a class="widget uib_w_1 d-margins" data-uib="jquery_mobile/button" data-role="button" id="btnHelloWorld">Hello World Button</a>
  <div class="widget uib_w_2" data-uib="media/text" id="txtHelloWorld">
    <div class="widget-container left-receptacle"></div>
    <div class="widget-container right-receptacle"></div>
    <div>
      <p>Aplikasi pertamaku di Intel XDK.</p>
    </div>
  </div><span class="uib_shim"></span>
</div>
```

Jika sulit menemukannya, intinya cari baris kode program berikut:

```
<div class="widget-container content-area vertical-col">
```

Kode program yang kami maksud adalah kode program di dalam elemen `<div>` tersebut. Jika ada perbedaan sedikit, tidak masalah, mungkin terdapat perbedaan saat Intel XDK menggenerate kode program kita.

Pada baris-baris kode program di bawah, coba lihat bagian kode program:

```
<a class="widget uib_w_1 d-margins" data-uib="jquery_mobile/button" data-role="button" id="btnHelloWorld">Hello World Button</a>
```



Itu adalah baris dari kode yang digenerate saat kita membuat elemen **Button**. Pada bagian yang ditunjukkan tanda panah, itu adalah hasil generate kode dari atribut-atribut yang kita isikan saat membuat tombol. Terdapat atribut **Id** dan **Label**.

Lalu lihat juga pada baris kode program berikut:

```
<div class="widget uib_w_2" data-uib="media/text" id="txtHelloWorld">
  <div class="widget-container left-receptacle"></div>
  <div class="widget-container right-receptacle"></div>
  <div>
    <p>Aplikasi pertamaku di Intel XDK.</p>
  </div>
</div><span class="uib_shim"></span>
```

Bisa menebak dan menemukan sendiri elemen apakah itu? Ya, itu adalah elemen **Text** dengan atribut **id** "txtHelloWorld" dan **Text** "Aplikasi pertamaku di Intel XDK".

Kesimpulannya, saat kita melakukan layouting dengan App Designer, sebenarnya kita sedang mempersingkat langkah kerja kita dengan membiarkan Intel XDK menggenerate sendiri kode HTML untuk kita.

Terdapat satu kesimpulan lain yang didapat: Pada pengembangan aplikasi mobile di Intel XDK atau aplikasi HTML5 lainnya, bahasa yang digunakan untuk layouting adalah bahasa HTML.

Sekarang, mari lanjut ke tahap pemberian perintah aksi program. Silakan cari baris kode program berikut ini:

```
25 <script type="text/javascript">
26   /* This function runs once the page is loaded, but appMobi is not yet active */
27   var init = function(){
28
29   };
30   window.addEventListener("load",init,false);
31
32   /* This code prevents users from dragging the page */
33   var preventDefaultScroll = function(event) {
34     event.preventDefault();
35     window.scroll(0,0);
36     return false;
37   };
38   document.addEventListener('touchmove', preventDefaultScroll, false);
39
40   /* This code is used to run as soon as appMobi activates */
41   var onDeviceReady=function(){
42
43     //hide splash screen
44     AppMobi.device.hideSplashScreen();
45   };
46   document.addEventListener("appMobi.device.ready",onDeviceReady,false);
47
48
49 </script>
```

Terlihat pada angka baris kode program pada gambar di atas, terlihat bahwa baris kode program tersebut berada pada baris 25-49. Atau mungkin beda-beda tipis pada baris kode program milik Anda. Itu adalah sebuah kode program dengan bahasa JavaScript. Pada pengembangan aplikasi menggunakan Intel XDK atau aplikasi HTML5 lainnya, kode perintah untuk aksi program menggunakan bahasa **JavaScript**. Kode di atas adalah kode-kode perintah dasar yang otomatis digenerate oleh Intel XDK.

Pada pengembangan aplikasi menggunakan Intel XDK atau aplikasi HTML5 lainnya, kode perintah untuk aksi program menggunakan bahasa JavaScript.

Sekarang kita akan menambahkan sebuah kode untuk perintah aksi program yang baru, kita akan simpan kode tersebut di bagian bawah sebelum tag tutup elemen `</script>` yaitu seperti ditunjukkan oleh panah pada gambar di bawah ini:

```
46     document.addEventListener("appMobi.device.ready",onDeviceReady,false);
47
48     | ←────────────────────────────────────────────────────────────────────────────────
49     </script>
```

Baris kode programnya adalah sebagai berikut, silakan diketikkan pada bagian tersebut:

```
function showHelloWorld(){
    $("#txtHelloWorld").text("Hello world Intel XDK");
}
```

Sehingga tampilan pada baris kode program adalah seperti berikut:

```
45     };
46     document.addEventListener("appMobi.device.ready",onDeviceReady,false);
47
48     function showHelloWorld(){
49         $("#txtHelloWorld").text("Hello World Intel XDK");
50     }
51
52     </script>
```

Mungkin kebanyakan dari Anda belum familiar dengan maksud dari kode program tersebut. Kode tersebut merupakan bahasa JavaScript dengan tambahan library JQuery Mobile (yang telah kita pilih di awal pengembangan aplikasi). Kode tersebut merupakan sebuah fungsi bernama **showHelloWorld()** yang saat dipanggil akan mengganti atribut **text** menjadi "Hello World Intel XDK" dari sebuah elemen pada aplikasi yang memiliki id "txtHelloWorld". Di mana elemen pada aplikasi kita yang memiliki id tersebut adalah **Text** yang kita buat sebelumnya. Singkatnya, fungsi tersebut ketika dipanggil akan membuat elemen **Text** yang kita buat sebelumnya berubah atribut **Text** atau isi teksnya menjadi "Hello World Intel XDK".

Berikutnya untuk dapat dijalankan, fungsi **showHelloWorld** tersebut harus dipanggil terlebih dahulu. Kita akan memanggil fungsi tersebut ketika tombol Hello World yang telah kita buat

sebelumnya disentuh (menggunakan jari pada smartphone atau diklik jika pada komputer PC).

Caranya, pertama-tama temukan baris kode program yang merupakan baris kode program yang telah digenerate Intel XDK untuk tombol yang telah kita buat, yaitu baris kode program berikut yang telah kita bahas di atas saat pertama kita memasuki tampilan Editor kode program:

```
<a class="widget uib_w_1 d-margins" data-uib="jquery_mobile/button" data-role="button" id="btnHelloWorld">Hello World Button</a>
```




Tambahkan kode program baru pada elemen `<a>` sebelum penutup tag elemen berupa tanda kurung siku tutup seperti yang ditunjukkan pada panah pada gambar di atas. Kode programnya adalah sebagai berikut:

```
ontouchstart="showHelloWorld() "
```

Sehingga baris kode program untuk tombol tersebut adalah sebagai berikut:

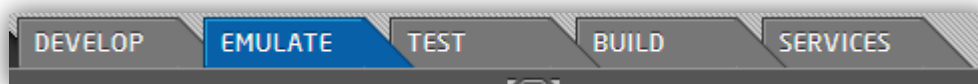
```
71         <div class="widget-container content-area vertical-col">
72         <a class="widget uib_w_1 d-margins" data-uib="jquery_mobile/button" data-role="button" id="btnHelloWorld"
73           ontouchstart="showHelloWorld()">Hello World Button</a>
           <div class="widget uib_w_2" data-uib="media/text" id="txtHelloWorld">
```



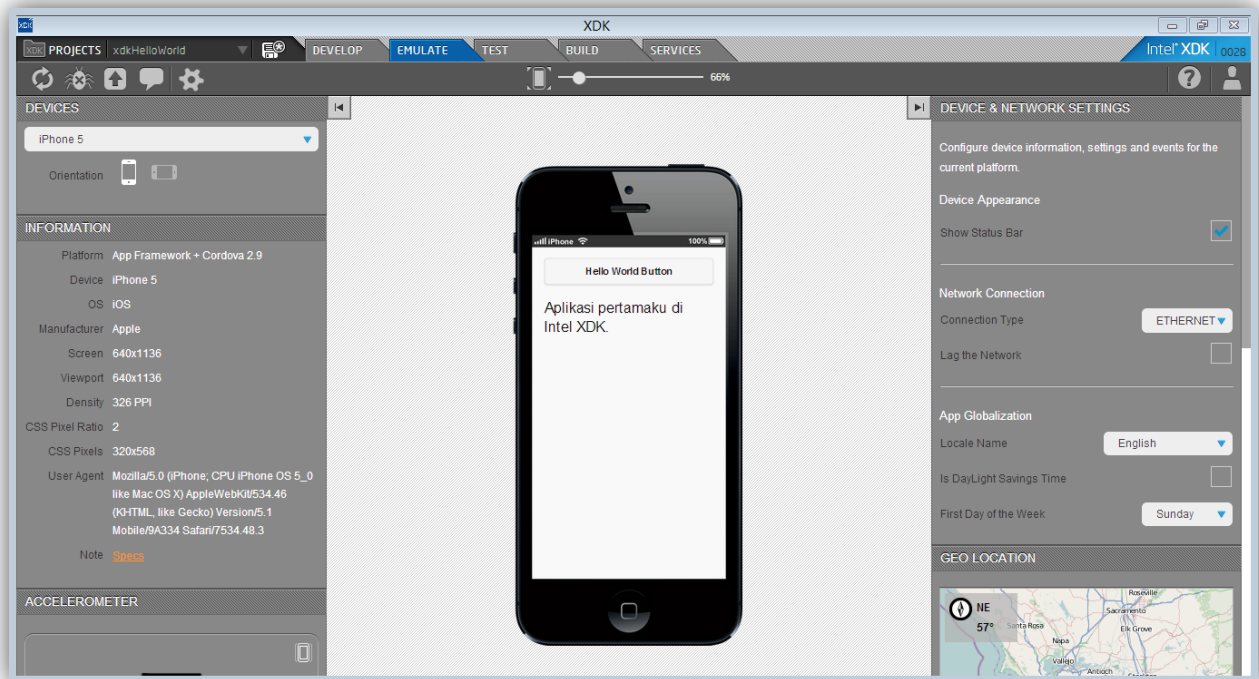
Terlihat sekarang bahwa selain atribut `class`, `data-uib`, `data-role`, dan `id`, terdapat satu atribut baru yaitu **ontouchstart**. Atribut ini berfungsi untuk mendefinisikan bahwa saat pada elemen tersebut mendapatkan input berupa touch pada device nanti (pada komputer PC berupa klik), maka akan memanggil fungsi `showHelloWorld()` yang telah kita buat dan definisikan sebelumnya.

4. Meng-emulate aplikasi

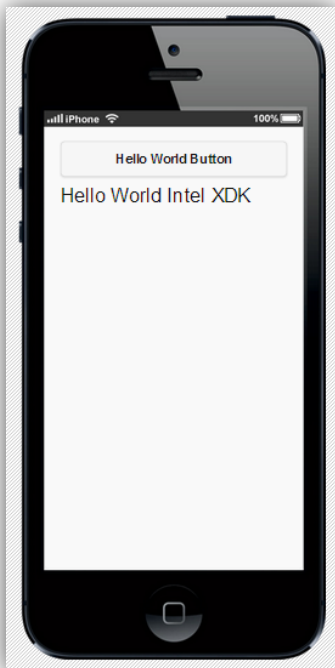
Sekarang mari kita jalankan kembali aplikasi kita, seperti sebelumnya, klik tombol Emulate di menu bagian atas.



Maka tampilan akan kembali dialihkan ke tampilan Emulate seperti berikut:



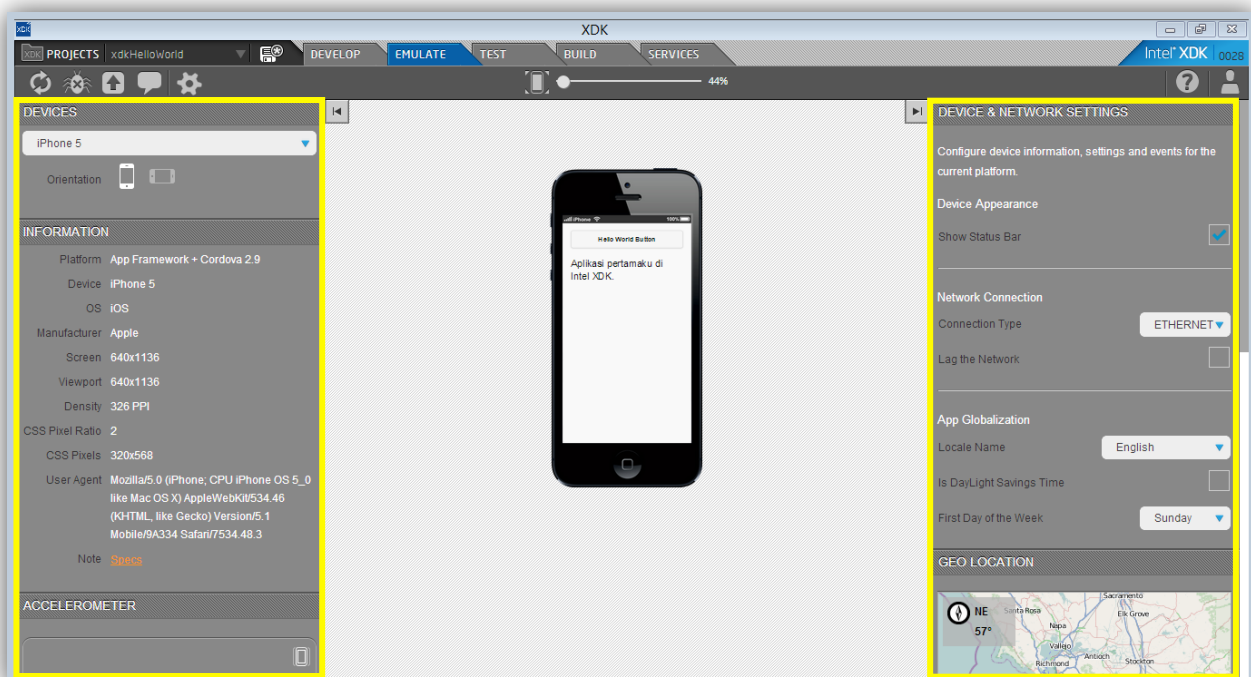
Sekarang, silakan coba klik tombol bertuliskan **"Hello World Button"** pada aplikasi kita. Seharusnya ketika tombol tersebut diklik akan merubah teks di bawahnya menjadi bertuliskan **"Hello World Intel XDK"**.



Pada menu di bagian atas, terdapat tombol **reload** aplikasi dan **slider** untuk zoom in dan zoom out device. Klik tombol **reload** aplikasi untuk mereload aplikasi yang sedang diemulate untuk kembali ke kondisi awal dan gunakan tombol **slider zoom in dan zoom out** untuk mengatur pembesaran tampilan device.



Dan pada bilah sebelah kiri dan kanan pada tampilan emulate terdapat beberapa fungsi untuk membantu kita meng-emulate aplikasi kita. Terdapat menu Device untuk mengganti pilihan device yang digunakan untuk meng-emulate aplikasi kita. Terdapat juga menu Accelerometer, Device & Network Settings, dan Geo Location. Itu semua untuk membantu kita dalam meng-emulate aplikasi kita dalam berbagai kemungkinan dan kondisi.



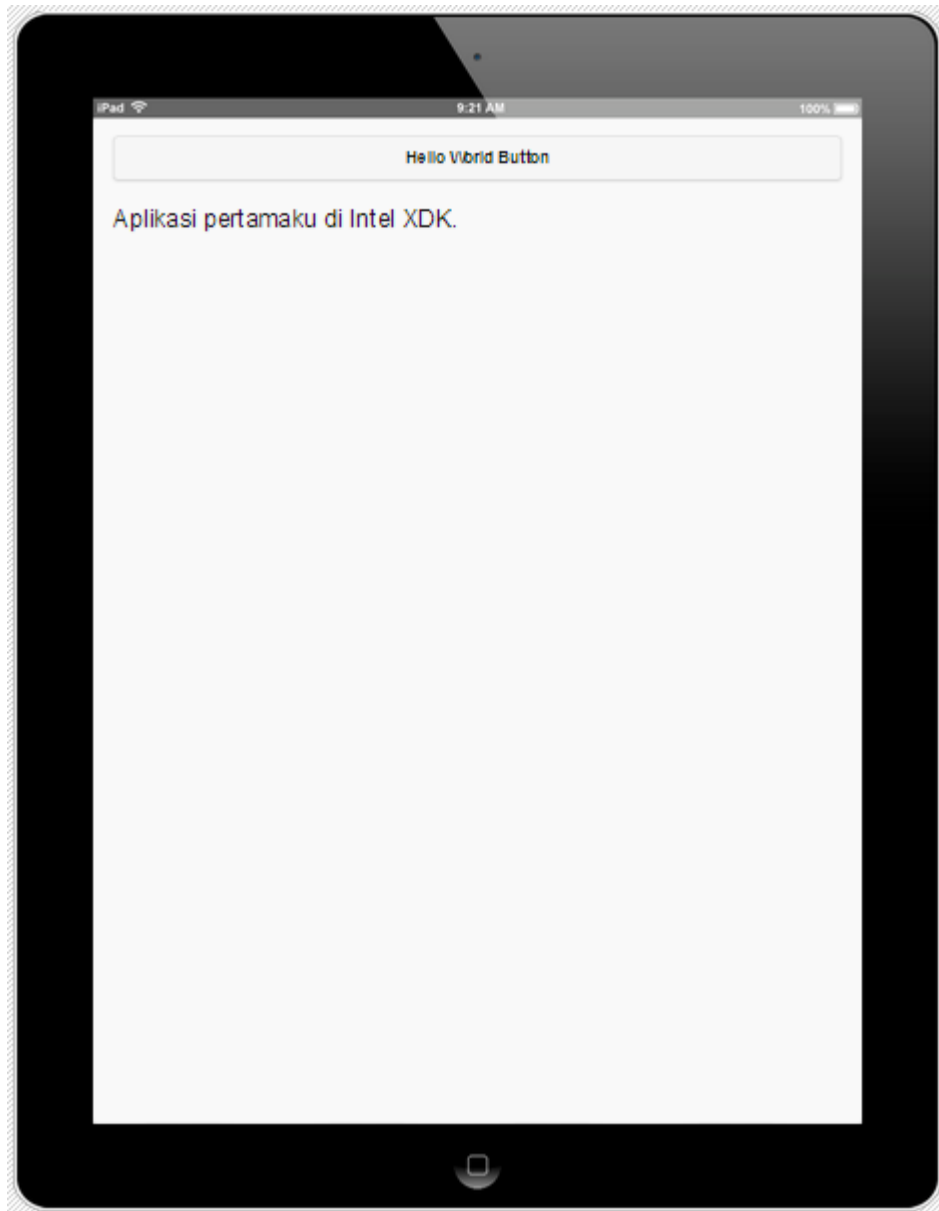
Sekarang silakan mencoba mengganti device yang kita gunakan untuk meng-emulate aplikasi Hello World kita. Pada bilah sebelah kiri, bagian menu Device terdapat tombol dropdown yang berisi list device yang dapat digunakan. Silakan diklik dan pilih device yang Anda inginkan.



Berikut ini tampilan aplikasi kita di Nokia Lumia 920:



Bahkan di sebuah iPad:



Lumayan bukan? 😊